(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 813 159 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
17.11.2004 Bulletin 2004/47

(51) Int Cl.[7]: **G06F 17/30**

(21) Application number: **97303915.9**

(22) Date of filing: **06.06.1997**

(54) **Method and system for prioritised downloading of embedded web objects**

Verfahren und System zum vorrangigen Herunterladen von eingebetteten Netzobjekten

Procédé et dispositif pour le téléchargement prioritoire d'objets web encastrés

(84) Designated Contracting States:
DE FR GB NL SE

(30) Priority: **10.06.1996 US 664048**

(43) Date of publication of application:
17.12.1997 Bulletin 1997/51

(73) Proprietor: **SUN MICROSYSTEMS, INC.**
Mountain View, California 94043-1100 (US)

(72) Inventor: **Nielsen, Jakob**
Atherton, California 94027 (US)

(74) Representative: **Read, Matthew Charles et al**
Venner Shipley LLP
20 Little Britain
London EC1A 7DH (GB)

(56) References cited:
EP-A- 0 584 029
EP-A- 0 632 671
US-A- 5 347 632

EP 0 813 159 B1

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

[0001] The present invention relates to an improved method and system for retrieving information elements in a computer network.

Background of the Invention

[0002] Many users access the World Wide Web through low-bandwidth connections, resulting in slow receipt of web pages (i.e., "the response time problem"). Even though bandwidth is expected to grow in coming years, the file size of multimedia and virtual reality objects will most likely grow at least as fast as the increase in bandwidth. Therefore, from a user's perspective, the response time problem will not go away.

[0003] Current web browsers exacerbate the response time problem because they retrieve the objects on a web page in the sequence in which they are listed in a web file that defines the web page (e.g., an HTML file). Since the objects in most web files are not sequentially ordered within the file on the basis of their relative importance to the other objects in the file, current web browsers will not retrieve web objects in the order of their relative importance. For example, a web page often starts with a headerbar that is comparatively unimportant and should be downloaded last. Using today's browsers, however, the headerbar would instead be downloaded first.

[0004] In US5347632, an interactive computer system is described, in which a central host serves a number of user terminals. A method of operating the system is disclosed which lessens the demand placed by users on the central host by preloading files on local terminals before the network is put into operation. When the network is in use, a number of the user's requests may be met using the pre-loaded information stored on their local terminal, reducing the need to download information from the host. A number of information elements, or objects, are defined in US5347632, one of which, an "external reference segment", includes a priority attribute to indicate the importance of the file for determining whether it should be pre-loaded onto the user terminals.

[0005] Embodiments of the present invention offer a more flexible approach to ameliorating the response time problem by downloading web objects based on a priority attribute associated with each object reference in the web file.

Summary of the Invention

[0006] According to the present invention, a method executed in a computer network for retrieving information elements, the computer network including a client computer and a server computer, the method comprising the steps of on the client computer: building a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element, sorting the list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and retrieving from the server computer to the client computer, the referenced information elements in ranked order.

[0007] According to the invention there is also provided a computer program product for retrieving information elements, in a computer network which includes a client computer and a server computer, the computer program product comprising code residing on a tangible medium that, when executed, causes a processor to: build a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element; sort the list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and retrieve from the server computer to the client computer, the referenced information elements in ranked order.

[0008] According to the invention, there is further provided an apparatus for retrieving information elements in a computer network, the computer network including a client computer and a server computer, the apparatus comprising: a mechanism configured to build a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element; a mechanism configured to sort a list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and a mechanism configured to retrieve from the server computer to the client computer, the referenced information elements in ranked order.

[0009] In a preferred embodiment, the information elements are web objects referenced in an HTML file. For example, a web object may be, but is not limited to, text, a graphical user interface element, an image file, an audio file, an applet, or other computer code. "Acting on" the information element typically includes, but is not limited to, displaying the text, displaying the graphical user interface element, displaying the image file, playing the audio file, executing the applet, or executing other computer code.

[0010] In a preferred embodiment of the invention, the method retrieves a web file and sorts one or more web object references according to a priority attribute associated with each web object reference. After ranking the web object references by priority, the method then retrieves each web object in the order that the references were ranked. After receiving a web object, a user acts upon the web object in the appropriate manner. By using

the steps of the preferred method, more important objects are retrieved before less important objects, thus allowing the user to act upon the more important objects sooner than the user could have acted on the important objects using methods available in the prior art.

[0011] In another embodiment, the web object references are sorted using a two-step process. In the first step, the web object references are ordered into an initial list and are assigned a sequence number according to the sequence of their appearance in the file. The list is then reordered by descending priority level as a primary sort key and by ascending sequence number as a secondary sort key. In this way objects with a higher priority will be sorted to the top of the list and objects with the same priority will be sorted such that the objects references early in the web file are sorted above those objects referenced later in the file.

[0012] In yet another embodiment, the web objects are retrieved in parallel in order to decrease retrieval time. First, the method determines whether at least one web object is currently being retrieved. If an object is currently being retrieved then the following steps are preferably followed to facilitate parallel retrieval of another web object. The method obtains an indication of an available rate of incoming bandwidth to the client computer and also obtains an indication of an available rate of outgoing bandwidth to the server computer storing the web object. The method then determines a minimum rate of the available incoming bandwidth and the available outgoing bandwidth. The method then accepts data associated with the next information element at a rate corresponding to a selected increment over the minimum rate. In this way, the overall rate of retrieval is increased.

Brief Description Of The Drawings

[0013]

Figure 1 is a block diagram of a computer network within which embodiments of the present invention can operate.

Figure 2 is an example of a series of HTML document tags forming a template for a typical hypertext document.

Figure 3 summarizes, in a table, the HTML tags used to create the HTML template document.

Figure 4 illustrates a hypertext document on Thomas Jefferson with a hot link for "the American Constitution".

Figure 5 illustrates the primary components of a Universal Resource Locator ("URL").

Figure 6 is a flow diagram which illustrates the preferred steps for retrieving and acting on information elements referenced in a file.

Figure 7 is a flow diagram which illustrates the preferred steps for retrieving remotely stored web objects.

Figure 8 illustrates the preferred steps for retrieving web objects in parallel.

Detailed Description

[0014] In an embodiment of the present invention, the method retrieves a file and sorts one or more information element references according to a priority attribute associated with each reference. After ranking the information element references by priority, the method then retrieves each information element in the order that their references were ranked. After receiving an information element, a user acts upon the information element in the appropriate manner. Typical information elements, and their actions, would include displaying an image file, executing an applet, or playing an audio file. By using the steps of the embodied method, more important information elements are retrieved before less important information elements, thus allowing the user to act upon the more important information elements sooner than the user could have acted on the important information elements using methods available in the prior art.

[0015] For purposes of the embodiment, the file will be referred to as a web file and the information elements will be referred to as objects or web objects. Those of ordinary skill will understand that the invention is not, however, limited to use with web files and web objects.

[0016] Figure 1 is a block diagram of a computer network within which embodiments of the present invention can operate. Most computer networks in use today are generally of the structure shown in Figure 1. Computer system 100 includes a processor 102 which fetches computer instructions from a primary storage 104 or a cache 105 through a bus 106 and executes those computer instruction. In executing computer instructions fetched from primary storage 104, processor 102 can retrieve data from or write data to primary storage 104, display information on one or more computer display devices 120, receive command signals from one or more user-input devices 130, or transfer data to other computer systems which collectively form a computer network (not shown). Processor 102 can be, for example, any of the SPARC processors available form Sun Microsystems, Inc. of Mountain View, California or any processors compatible therewith, primary storage 104 can include any type of computer primary storage including, without limitation, randomly accessible memory (RAM), read-only memory (ROM), and storage devices which include magnetic and optical storage media such as magnetic or optical disks. Computer display devices 120 can include, for example, printers and computer display screens such as cathode-ray tubes (CRTs), light-emitting diode (LED) displays, and liquid crystal displays (LCDs). User-input devices 130 can include without limitation electronic keyboards and pointing devices such as electronic mice, trackballs, lightpens, thumbwheels, digitizing tablets, and touch sensitive pads.

[0017] Computer system 100 can be, e.g., any of the

SPARCstation workstation computer systems available form Sun Microsystems, Inc. of Mountain View, California, any other Macintosh computer systems based on the PowerPC processor and available from Apple Computers, Inc. of Cuptertino, California, or any computer system compatible with the IBM PC computer systems available form International Business Machines, Corp of Somers, New York, which are based on the X86 series of processors available from Intel Corporation or compatible processors. Sun, Sun Microsystems, and the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

[0018] Also executing within processor 102 from primary storage 104 is a runtime environment 112. Runtime environment 112 is generally a set of computer programs which enable computer system 100 to understand and process commands, control input and output of computer system 100 through user-input devices 130 and computer display devices 120, schedule computer processes for execution, manage data stored in various storage elements of primary storage 104 of computer system 100, and control the operation of peripheral devices (such as secondary storage 140) coupled to computer system 100. In some embodiments, runtime environment 112 may be an operating system or an operating system with a kernel. The preferred operating system is the Solaris operating system from Sun Microsystems, Inc., of Mountain View, California.[1] The kernel of an operating system is that portion of the operating system which manages the interface between computer processes (e.g., web browser 108) and user-input devices 130 and computer display devices 120, manages primary storage 104, schedules computer processes for execution, and maintains a file system which in turn manages storage of data (e.g., web file 110) on various storage elements of primary storage 104.

[0019] The preferred web file 110 is written in HTML, although any language which supports hypertext could be used. A brief discussion of HTML may lay the foundation for a clearer understanding of the teachings and suggestions of the present invention. The HTML markup language is analogous in some ways to the formatting codes used in word processing documents. A word processing document viewed through a word processing program is actually a combination of the text that you see and a series of hidden formatting codes (e.g., the carriage return, bold, and underline codes) which instruct the word processing program to display the word processing document in a specified way. Similarly, a hypertext document is actually a combination of the text that you see and a series of hidden "tags" or "anchors" (e.g., tags for new paragraphs, graphic images, hypertext links, etc.) which instruct the browser program to display the hypertext document in a specified way.

1. Sun and Solaris are trademarks or registered trademarks of Sun Microsystems,Inc., in the United States and other countries.

[0020] A hypertext document is usually broken down into sections, with each section delineated by one or more HTML tags. HTML tags are formatting codes surrounded by the characters "<" and ">" (i.e., less than and greater than symbols). Some HTML tags have a start tag and an end tag. In general, end tags are in the format </"symbol"> where the "symbol" is the character string found between the characters < and > in the start tag. Figure 2 is an example of a series of HTML document tags forming a template for a typical hypertext document. For example, the document of Figure 2 is defined as an HTML document using the tags <html> and </html>. Then the "head" to the document, which typically includes a title, is defined using the tags <head>, </head>, <title>, and </title>, respectively. Following the head comes the "body" of the document which is often organized into subtopics with different levels of headings. The body is defined by the tags <body> and </body>. Headings are indicated by the tags <h#> and </h#>, where # is the level of the heading. Heading levels indicate the relative size of the heading. Heading level 1 is the largest heading size and heading level 6 is the smallest heading size. Finally, it is good practice to indicate the author of the document at the bottom of the document using the tags <address> and </address>. Figure 3 summarizes, in a table, the HTML tags used to create the HTML template document.

[0021] Once the HTML template has been established, text is added to create a basic hypertext document. In order to improve readability, the author adds HTML character and paragraph formatting tags to the document. For example, the <p> tag instructs the browser to begin a new paragraph. If an author wants to highlight some text in bold, the author inserts the tag at the beginning of the text to be highlighted and inserts a tag at the end of the text to be highlighted. The tags and indicate text to display in italics.

[0022] If HTML was merely made up of the document, paragraph, and character formatting tags discussed above, it would only allow an author to define a document which stands by itself. Fortunately, additional HTML tags allow an author to "link" documents together. If a reader of a hypertext document wants to know more about a topic before reading the rest of the current hypertext document, the reader selects a "link" or "hot link", which retrieves and displays a new document that provides related information. Figure 4 illustrates a hypertext document (i.e, a "source document") on Thomas Jefferson with a hot link named "the American Constitution". The link could take the reader to a second hypertext document (i.e., a "destination document") which, for example, displays the text of the American Constitution or which provides more information on Thomas Jefferson's role in the drafting of the American Constitution.

[0023] In HTML, a hot link to a destination document

is made by placing a "reference anchor" around the text to be highlighted (e.g., "the American Constitution") and then providing a network location where the destination document is located. Reference anchors extend the idea of start and end tags. A reference anchor is created when the start tag <a> and the end tag are placed around the text to be highlighted (e.g., <a> the American Constitution). Then attribute information that identifies the network location of the destination document is inserted within the <a> reference tag. In HTML, the "href=" attribute, followed by the network location for the destination document, is inserted within the <a> tag. For example,

 the American Constitution illustrates the basic format for a reference anchor. On the web, network locations of hypertext documents are provided using the Universal Resource Locator ("URL") naming scheme.

[0024] Figure 5 illustrates the primary components of a URL. A service type 501 is a required part of a URL. The service type tells the user's browser how to contact the server for the requested data. The most common service type is the HyperText Transport Protocol or http. The web can handle several other services including gopher, wais, ftp, netnews, and telnet and can be extended to handle new service types. A system name 503 is also a required part of a URL. The system name is the fully qualified domain name of the server which stores the data being requested. A port 505 ia an optional part of a URL. Ports are the network socket addresses for specific protocols. By default, http connects at port 80. Ports are only needed when the server does not communicate on the default port for that service. A directory path 507 is a required part of a URL. Once connected to the system in question, a path to the file must be specified. A filename 509 is an optional part of a URL. The file name is the data file itself. The server can be configured so that if a filename isn't specified, a default file or directory listing is returned. A search component 511 is another optional part of a URL. If the URL is a request to search a data base, the query can be embedded in the URL. The search component is the text after the ? or # in a URL.

[0025] Substituting the URL "http://system/dir/file.html" into the example above, the reference anchor:

 the American Constitution

identifies an html file to retrieve and display when a user selects "the American Constitution" hot link.

[0026] As is described in more detail below, embodiments of the present invention extend the idea of anchors and tags by inserting a "PRIORITY" attribute into existing anchor and tag formats. For example, traditionally, an author specifies the downloading of an embedded object by including HTML markup of the following type in the HTML code specifying the page: <IMG SRC=mypicture.gif>. Using the present invention, the author assigns priority to the downloading of the embedded objects by adding a PRIORITY attribute to the tag for the object: <IMG SRC=mypicture.gif PRIORITY=5>.

[0027] The PRIORITY attribute should be set equal to a number (either negative, positive, or zero). If the value of a PRIORITY attribute is not a number then the browser 108 assumes that the priority of that embedded object is zero.

[0028] If a page using priority tags is displayed by a browser that has not yet been enhanced to work with the present invention, the PRIORITY attributes will simply be ignored, and the embedded objects will be downloaded following the standard rules used by that browser.

[0029] If a page is displayed by a browser that has been enhanced to work with embodiments of the present invention, the embedded objects will preferably be downloaded following the steps illustrated in Figures 6, 7, and 8, and described in detail below.

[0030] Continuing with the discussion of Figure 1, computer network 10 also includes a network connection 145 for facilitating communication between host computer system 100 and server(s) 150. Network connection 145 can be any well know mechanism for facilitating communication between computers, such as, without limitation, a local area network, a wide area network, the Internet, or any of the well known wireless communication systems. Server(s) 150 typically store the information elements that are referenced in the web file 110 and are retrieved using the teachings and suggestions of the present invention.

Flow Diagrams Of The Preferred Embodiment

[0031] Figure 6 is a flow diagram which illustrates the preferred steps for retrieving and acting on web objects referenced in a web file. In step 601, the browser reads the web file and displays the text on the screen following the well-established prior art methods for web-page display. This method involves leaving space on the page for embedded objects that need space (typically images and applets). If the object has WIDTH and HEIGHT attributes, then the amount of space reserved for the object is that specified by these attributes and otherwise, the space is initially set equal to the size of an icon (typically 32 by 32 pixels) and later adjusted to the actual size of the object when it has been downloaded (adjusting the space may involve moving subsequent parts of the page either up or down, depending on whether the initially reserved space was too large or too small).

[0032] In step 603, the browser determines if any web objects referenced by the web file are stored in its cache. If cached web objects exist then, in step 605, the browser determines if unprocessed cached web objects remain to be processed. If unprocessed web objects remain then, in step 607, the browser retrieves the next web object that is available in its cache. In step 609, the browser acts on the retrieved web object. Typically, "act-

ing on an object" means displaying it if it is an image file, playing it if it is an audio file, and executing it if it is an applet or computer code, but the exact action to take is dependent on the media type of the object and follows prior art for currently known object types.

[0033] After all cached web objects have been retrieved, the browser invokes steps to retrieve web objects stored on a remote server (step 611). Figure 7, which is described in more detail below, illustrates the preferred steps for retrieving remotely stored web objects. Upon receiving the remotely stored objects, the bowser acts upon the retrieved web object (step 613).

[0034] As was mentioned above, Figure 7 illustrates the preferred steps for retrieving remotely stored web objects. In step 701, the browser builds a list of all the embedded objects in the web file that were not found in the cache. The objects are preferably numbered by their sequential appearance in the web file (e.g., the first object is object 1, the second is object 2, etc.). If the object is referenced multiple times, it is preferably assigned the sequence number of its first reference in the file. Any object that does not have a PRIORITY attribute is preferably assigned a priority of zero.

[0035] In step 703, the browser preferably sorts the list by descending priority as the primary sort key and ascending sequence number as the secondary sort key. Thus, objects with a high priority will be sorted on top and objects with the same priority will be sorted such that the ones that are referenced early in the web file are sorted above those that are referenced later in the web file.

[0036] In step 705, the browser initiates retrieval of the objects from the remote server(s), preferably in the order in which they appear on the sorted list. Upon completion of step 705, processing ends in the method of Figure 7 but continues with step 613 of Figure 6 where the retrieved objects are acted upon (e.g., displayed, played, or executed).

[0037] Figure 8 illustrates the preferred steps for retrieving web objects in parallel. The steps of Figure 8 are performed as the preferred implementation of step 705 of Figure 7. Once retrieval of an object has started, the browser preferably determines whether to start retrieval of the next object on the list by the following rules:

(i) if no objects are currently being retrieved (step 801), then start retrieving the next object (step 803).
(ii) if objects are currently being retrieved (step 801), but there is available incoming bandwidth AND available bandwidth to the server holding the object (steps 805 & 807), then start retrieving the next object, but only accept incoming packets for that object at 10% more than the minimum of the available incoming bandwidth and the available bandwidth to the server holding the object, recalculating this number once every second (steps 809 & 811) until all packets for the next object have been retrieved (step 813)

Regarding the terms used:

[0038]

"available incoming bandwidth" is preferably defined as follows: total incoming bandwidth minus the sum of the number of bits received from higher ranked objects during the last second.
"total incoming bandwidth" is preferably defined as follows: every ten seconds, the browser adds up the number of bits it has received from any source outside the computer on which it is running during those ten seconds. The browser stores the last one hundred of these measures that have been larger than zero, and total incoming bandwidth in bits per second is the largest of these one hundred measures divided by ten. When the user quits the browser, the current total incoming bandwidth number
"available bandwidth to a server" is preferably defined as follows: total bandwidth to that server minus the sum of the number of bits received from higher ranked objects being retrieved from that server during the last second.
"total bandwidth to a server" is preferably defined as follows: every two seconds, the browser adds up the number of bits it has received from that server during those two seconds. The browser stores the last twenty of these measures that have been larger than zero, and total incoming bandwidth in bits per second is the largest of these twenty measures divided by two. When the user quits the browser, the current total bandwidths to the servers are stored in a preference file with one entry for each server that has been contacted during the current session. The first time the browser accesses a server during a session, it reads the previous total bandwidth number for that server from the preference file and uses that number as the total bandwidth to the server until the first measure is available from the current session. The very first time the browser accesses a server (or if the preference file cannot be found), the initial value for total bandwidth to the server is set equal to 28,800 bps.

[0039] While specific embodiments have been described herein for purposes of illustration, various modifications may be made without departing from the scope of the invention. Accordingly, the invention is not limited to the above described embodiments. Instead the invention is defined by the appended claims.

Claims

1. A method executed in a computer network (10) for retrieving information elements, the computer network including a client computer (100) and a server computer (150), the method comprising the steps

of on the client computer (100):

building a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element;
sorting the list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and
retrieving from the server computer to the client computer, the referenced information elements in the ranked order.

2. A method according to claim 1, wherein the steps of building and sorting the list of information element references further comprise the steps of:

receiving a file (110) which stores the information element references;
ordering the information element references into an initial list according to the information element reference's sequential appearance in the file, the sequential appearance having an association with one or more sequence numbers; and
reordering the initial list by descending priority level as a primary sort key and by ascending sequence number as a secondary sort key.

3. A method according to claim 1 or claim 2, wherein the step of retrieving the referenced information elements further comprises the steps of:

determining whether at least one information element referenced by the list is currently being retrieved;
when the determination indicates that at least one information element is currently being retrieved, performing the following steps to facilitate retrieval of a next information element,
obtaining an indication of an available rate of incoming bandwidth to the client computer (100);
obtaining an indication of an available rate of outgoing bandwidth to the server computer (150);
determining a minimum rate of the available incoming bandwidth and the available outgoing bandwidth; and
accepting data associated with the next information element at a rate corresponding to an increment over the minimum rate.

4. A method according to claim 3, wherein the increment is 10%.

5. A method according to claim 3, wherein the available incoming bandwidth is a total incoming bandwidth minus the sum of the number of bits received from higher ranked information elements running during a previous time period.

6. A method according to claim 5, wherein the total incoming bandwidth is, within a time period, defined by,

determining a number of bits received from any source outside the client computer (100);
comparing the determined number of bits with the quantity of bits received over periods of time other than said time period; and
selecting the largest number of bits determined from the comparison as the total incoming bandwidth.

7. A method according to claim 3 or claim 4, wherein the available bandwidth to the server computer (150) storing the next information element is the total bandwidth to the server computer minus a sum of the number of bits received by the client computer (100), during a period, from retrieving from the server computer an information element ranked higher than the next information element.

8. A method according to any preceding claim, further comprising the step of acting on each information element after it is retrieved.

9. A method according to claim 8, wherein the step of acting is one of displaying text, displaying a graphical user interface element, displaying an image file, playing an audio file, executing an applet, and executing a plurality of computer code.

10. A method according to any preceding claim, wherein the information element is one of a text, a graphical user interface element, an image file, and audio file, an applet, and a plurality of computer code.

11. A computer program product for retrieving information elements, in a computer network (10) which includes a client computer (100) and a server computer (150), the computer program product comprising code residing on a tangible medium that, when executed, causes a processor (102) to:

build a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element;
sort the list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and

retrieve from the server computer to the client computer, the referenced information elements in the ranked order.

12. An apparatus for retrieving information elements in a computer network (10), the computer network including a client computer (100) and a server computer (150), the apparatus comprising:

a mechanism configured to build a list of information element references, at least one of the information element references having a priority level associated with the corresponding information element;
a mechanism configured to sort a list of information element references so that the list is ranked from a first information element reference with a highest priority to a last information element reference with a lowest priority; and
a mechanism configured to retrieve from the server computer to the client computer, the referenced information elements in the ranked order.

13. An apparatus according to claim 12, wherein the mechanisms configured to build and sort the list of information element references further comprise:

a mechanism configured to receive a file (110) which stores the information elements;
a mechanism configured to order the information element references into an initial list according to the information element reference's sequential appearance in the file, the sequential appearance having an association with one or more sequence numbers; and
a mechanism configured to reorder the initial list by descending priority level as a primary sort key and by ascending sequence number as a secondary sort key.

14. An apparatus according to claim 12 or 13, wherein the mechanism configured to retrieve the information elements further comprises:

a mechanism configured to determine whether at least one information element referenced in the list is currently being retrieved;
a mechanism configured to facilitate retrieval of a next information element referenced in the list, when the determination indicates that at least one information element is currently being retrieved, comprising a mechanism configured to obtain an indication of an available rate of incoming bandwidth to the client computer (100), a mechanism configured to obtain an indication of an available rate of outgoing bandwidth to the server computer (150), a mechanism configured to determine a minimum rate of the available incoming bandwidth and the available outgoing bandwidth and a mechanism configured to accept data associated with the next information element at a rate corresponding to an increment over the minimum rate.

15. An apparatus according to claim 14, wherein the increment is 10%.

16. An apparatus according to claim 14 or 15, wherein the available incoming bandwidth is a total incoming bandwidth minus the sum of the number of bits received from higher ranked information elements running during a previous time period.

17. An apparatus according to any one of claims 14, 15 or 16, wherein the mechanism configured to obtain the total incoming bandwidth within a given time period, includes:

a mechanism configured to determine a number of bits received from any source outside the computer system associated with the display device;
a mechanism configured to compare the determined number of bits with the quantity of bits received over periods of time other than the given periods of time; and
a mechanism configured to select the largest number of bits determined from the comparison as the total incoming bandwidth.

18. An apparatus according to any one of claims 14 to 17, wherein the mechanism configured to obtain the available bandwidth to the computer system storing the next information element includes a mechanism configured to determine the total bandwidth to the computer system storing the next information element minus a sum of the number of bits received, during a given period, from retrieving an information element ranked higher than the next information element.

19. An apparatus according to any one of claims 12 to 18, further comprising a mechanism configured to act on each information element after it is retrieved.

20. An apparatus according to claim 19, wherein the mechanism configured to act performs at least one of the following acts: displays text, or displays a graphical user element, or displays an image file, or plays an audio file, or executes an applet, or executes a plurality of computer code.

21. An apparatus according to any one of claims 12 to 20, wherein the information element is one of a text,

a graphical user interface element, an image file, an audio file, an applet, and a plurality of computer code.

Patentansprüche

1. In einem Computernetz (10) ausgeführtes Verfahren zum Abrufen von Informationselementen, wobei das Computernetz einen Client-Computer (100) und einen Server-Computer (150) enthält, wobei das Verfahren die folgenden Schritte auf dem Client-Computer (100) umfasst:

   Aufstellen einer Liste von Informationselement-Referenzen, wobei mindestens eine der Informationselement-Referenzen eine dem entsprechenden Informationselement zugeordnete Prioritätsebene aufweist;
   Sortieren der Liste von Informationselement-Referenzen, so dass die Liste von einer ersten Informationselement-Referenz mit einer höchsten Priorität nach einer letzten Informationselement-Referenz mit einer niedrigsten Priorität geordnet ist; und
   Abrufen der referenzierten Informationselemente in der geordneten Reihenfolge von dem Server-Computer zu dem Client-Computer.

2. Verfahren nach Anspruch 1, bei dem die Schritte des Aufstellens und Sortierens der Liste von Informationselement-Referenzen weiterhin die folgenden Schritte umfassen:

   Empfangen einer Datei (110), die die Informationselement-Referenzen speichert; Ordnen der Informationselement-Referenzen in einer Anfangsliste in Übereinstimmung mit dem sequentiellen Auftreten der Informationselement-Referenzen in der Datei, wobei das sequentielle Auftreten eine Zuordnung zu einer oder mehreren Sequenznummem aufweist; und
   Neuordnen der Anfangsliste nach absteigender Prioritätsebene als ein primärer Sortierschlüssel und nach absteigender Sequenznummer als ein sekundärer Sortierschlüssel.

3. Verfahren nach Anspruch 1 oder 2, bei dem der Schritt des Abrufens der referenzierten Informationselemente weiterhin die folgenden Schritte umfasst:

   Bestimmen, ob mindestens ein durch die Liste referenziertes Informationselement gegenwärtig abgerufen wird;
   wenn die Bestimmung anzeigt, dass mindestens ein Informationselement gegenwärtig abgerufen wird, Durchführen der folgenden Schritte, um das Abrufen eines nächsten Informationselements zu erleichtern:
   Erhalten einer Anzeige einer verfügbaren Rate von am Client-Computer (100) ankommender Bandbreite;
   Erhalten einer Anzeige einer verfügbaren Rate von zum Server-Computer (150) abgehender Bandbreite;
   Bestimmen einer minimalen Rate der verfügbaren ankommenden Bandbreite und der verfügbaren abgehenden Bandbreite; und
   Annehmen der dem nächsten Informationselement zugeordneten Daten mit einer Rate, die einem Inkrement über der minimalen Rate entspricht.

4. Verfahren nach Anspruch 3, bei dem das Inkrement 10 % ist.

5. Verfahren nach Anspruch 3, bei dem die verfügbare ankommende Bandbreite eine gesamte ankommende Bandbreite minus der Summe der Zahl von Bits ist, die von höher eingestuften Informationselementen empfangen werden, die während einer vorhergehenden Zeitspanne laufen.

6. Verfahren nach Anspruch 5, bei dem die gesamte ankommende Bandbreite innerhalb einer Zeitspanne durch Folgendes definiert wird:

   Bestimmen einer Zahl von Bits, die von irgendeiner Quelle außerhalb des Client-Computers (100) empfangen werden;
   Vergleichen der Bestimmten Zahl von Bits mit der Menge von in anderen Zeitspannen als der genannten Zeitspanne empfangenen Bits; und
   Auswählen der aus dem Vergleich bestimmten größten Zahl von Bits als die gesamte ankommende Bandbreite.

7. Verfahren nach Anspruch 3 oder Anspruch 4, bei dem die verfügbare Bandbreite zum Server-Computer (150), der das nächste Informationselement speichert, die gesamte Bandbreite zum Server-Computer minus einer Summe der Zahl der vom Client-Computer (100) empfangenen Bits während einer Zeitspanne vom Abrufen eines höher als das nächste Informationselement eingestuften Informationselements vom Server-Computer ist.

8. Verfahren nach einem vorhergehenden Anspruch, das weiterhin den Schritt umfasst, auf jedes Informationselement zu agieren, nachdem es abgerufen ist.

9. Verfahren nach Anspruch 8, bei dem der Schritt des Agierens darin besteht, Text anzuzeigen, ein Grafikbenutzerschnittstellenelement anzuzeigen, eine

Bilddatei anzuzeigen, eine Audiodatei abzuspielen, ein Applet auszuführen oder eine Mehrzahl von Computercodes auszuführen.

10. Verfahren nach einem vorhergehenden Anspruch, bei dem das Informationselement ein Text, ein Grafikbenutzerschnittstellenelement, eine Bilddatei, eine Audiodatei, ein Applet oder eine Mehrzahl von Computercodes ist.

11. Computerprogrammerzeugnis zum Abrufen von Informationselementen in einem Computernetz (10), das einen Client-Computer (100) und einen Server-Computer (150) enthält, wobei das Computerprogrammerzeugnis Code umfasst, der sich auf einem greifbaren Medium befindet und bei Ausführung einen Prozessor (102) zu Folgendem veranlasst:

Aufstellen einer Liste von Informationselement-Referenzen, wobei mindestens eine der Informationselement-Referenzen eine dem entsprechenden Informationselement zugeordnete Prioritätsebene aufweist;
Sortieren der Liste von Informationselement-Referenzen, so dass die Liste von einer ersten Informationsetement-Referenz mit einer höchsten Priorität nach einer letzten Informationseiement-Referenz mit einer niedrigsten Priorität geordnet ist; und
Abrufen der referenzierten Informationselemente in der geordneten Reihenfolge von dem Server-Computer zu dem Client-Computer.

12. Vorrichtung zum Abrufen von Informationselementen in einem Computernetz (10), wobei das Computernetz (10) einen Client-Computer (100) und einen Server-Computer (150) enthält, wobei die Vorrichtung Folgendes umfasst:

einen Mechanismus, der konfiguriert ist, eine Liste von Informationselement-Referenzen aufzustellen, wobei mindestens eine der Informationselement-Referenzen eine dem entsprechenden Informationselement zugeordnete Prioritätsebene aufweist;
einen Mechanismus, der konfiguriert ist, eine Liste von Informationselement-Referenzen zu sortieren, so dass die Liste von einer ersten Informationselement-Referenz mit einer höchsten Priorität nach einer letzten Informationselement-Referenz mit einer niedrigsten Priorität geordnet ist; und
einen Mechanismus, der konfiguriert ist, die referenzierten Informationselemente in der geordneten Reihenfolge von dem Server-Computer zu dem Client-Computer abzurufen.

13. Vorrichtung nach Anspruch 12, bei der die Mechanismen, die konfiguriert sind, die Liste von Informationselement-Referenzen aufzustellen und zu sortieren, weiterhin Folgendes umfassen:

einen Mechanismus, der konfiguriert ist, eine Datei (110) zu empfangen, die die Informationselemente speichert,
einen Mechanismus, der konfiguriert ist, die Informationselement-Referenzen in Übereinstimmung mit dem sequentiellen Auftreten der Informationselement-Referenzen in der Datei in einer Anfangsliste zu ordnen, wobei das sequentielle Auftreten eine Zuordnung zu einer oder mehreren Sequenznummern aufweist; und
einen Mechanismus, der konfiguriert ist, die Anfangsliste nach absteigender Prioritätsebene als ein primärer Sortierschlüssel und nach absteigender Sequenznummer als ein sekundärer Sortierschlüssel neu zu ordnen.

14. Vorrichtung nach Anspruch 12 oder 13, bei der der Mechanismus, der konfiguriert ist, die Informationselemente abzurufen, weiterhin Folgendes umfasst:

einen Mechanismus, der konfiguriert ist, zu bestimmen, ob mindestens ein in der Liste referenziertes Informationselement gegenwärtig abgerufen wird;
einen Mechanismus, der konfiguriert ist, das Abrufen eines nächsten in der Liste referenzierten Informationselements zu erleichtern, wenn die Bestimmung anzeigt, dass mindestens ein Informationselement gegenwärtig abgerufen wird, mit einem Mechanismus, der konfiguriert ist, eine Anzeige einer verfügbaren Rate von am Client-Computer (100) ankommender Bandbreite zu erhalten, einem Mechanismus, der konfiguriert ist, eine Anzeige einer verfügbaren Rate von zum Server-Computer (150) abgehender Bandbreite zu erhalten, einem Mechanismus, der konfiguriert ist, eine minimale Rate der verfügbaren ankommenden Bandbreite und der verfügbaren abgehenden Bandbreite zu bestimmen, und einem Mechanismus, der konfiguriert ist, die dem nächsten Informationselement zugeordneten Daten mit einer Rate anzunehmen, die einem Inkrement über der minimalen Rate entspricht.

15. Vorrichtung nach Anspruch 14, bei der das Inkrement 10 % ist.

16. Vorrichtung nach Anspruch 14 oder 15, bei der die verfügbare ankommende Bandbreite eine gesamte ankommende Bandbreite minus der Summe der Zahl von Bits ist, die von höher eingestuften Infor-

mationselementen empfangen werden, die während einer vorhergehenden Zeitspanne laufen.

17. Vorrichtung nach einem der Ansprüche 14, 15 oder 16, bei der der Mechanismus, der konfiguriert ist, die gesamte ankommende Bandbreite innerhalb einer gegebenen Zeitspanne zu erhalten, Folgendes enthält:

einen Mechanismus, der konfiguriert ist, eine Zahl von Bits zu bestimmen, die von irgendeiner Quelle außerhalb des zu dem Anzeigegerät gehörenden Computersystems empfangen werden;
einen Mechanismus, der konfiguriert ist, die Bestimmte Zahl von Bits mit der Menge von in anderen Zeitspannen als den genannten Zeitspannen empfangenen Bits zu vergleichen; und
einen Mechanismus, der konfiguriert ist, die aus dem Vergleich bestimmte größte Zahl von Bits als die gesamte ankommende Bandbreite auszuwählen.

18. Vorrichtung nach einem der Ansprüche 14 bis 17, bei der der Mechanismus, der konfiguriert ist, die verfügbare Bandbreite für das Computersystem zu erhalten, das das nächste Informationselement speichert, einen Mechanismus enthält, der konfiguriert ist, die gesamte Bandbreite für das Computersystem, das das nächste Informationselement speichert, minus einer Summe der Zahl der während einer gegebenen Zeitspanne vom Abrufen eines höher als das nächste Informationselement eingestuften Informationselements empfangenen Bits zu bestimmen.

19. Vorrichtung nach einem der Ansprüche 12 bis 18, die weiterhin einen Mechanismus umfasst, der konfiguriert ist, auf jedes Informationselement zu agieren, nachdem es abgerufen ist.

20. Vorrichtung nach Anspruch 19, bei der der Mechanismus, der konfiguriert ist, zu agieren, mindestens eine der folgenden Aktionen durchführt: Text anzeigt, oder ein Grafikbenutzerelement anzeigt, oder eine Bilddatei anzeigt, oder eine Audiodatei abspielt, oder ein Applet ausführt, oder eine Mehrzahl von Computercodes ausführt.

21. Vorrichtung nach einem der Ansprüche 12 bis 20, bei der das Informationselement ein Text, ein Grafikbenutzerschnittstellenelement, eine Bilddatei, eine Audiodatei, ein Applet oder eine Mehrzahl von Computercodes ist.

Revendications

1. Procédé exécuté dans un réseau informatique (10) pour extraire des éléments d'information, le réseau informatique comprenant un ordinateur client (100) et un ordinateur serveur (150), le procédé comprenant les étapes consistant à, sur l'ordinateur client (100) :

construire une liste de références d'élément d'information, au moins une des références d'élément d'information possédant un niveau de priorité associé à l'élément d'information correspondant ;
trier la liste de références d'élément d'information, de manière que la liste soit rangée d'une première référence d'élément d'information possédant la plus haute priorité à une dernière référence d'élément d'information possédant la plus basse priorité ; et
extraire, de l'ordinateur serveur vers l'ordinateur client, les éléments d'information référencés dans l'ordre de rangement.

2. Procédé selon la revendication 1, dans lequel les étapes consistant à construire et trier la liste de références d'élément d'information comprend de plus les étapes consistant à :

recevoir un fichier (110) qui stocke les références d'élément d'information ;
ordonner les références d'élément d'information en une liste initiale selon l'apparition séquentielle dans le fichier de la référence d'élément d'information, l'apparition séquentielle étant associée à un ou plusieurs numéros de séquence ; et
réordonner la liste initiale par niveau de priorité décroissante comme clé de tri primaire et par numéro de séquence croissant comme clé de tri secondaire.

3. Procédé selon la revendication 1 ou la revendication 2, dans lequel l'étape consistant à extraire les éléments d'information référencés comprend de plus les étapes consistant à :

déterminer si au moins un élément d'information référencé par la liste est extrait à ce moment là ;
quand la détermination indique qu'au moins un élément d'information est extrait à ce moment là, réaliser les étapes suivantes pour faciliter l'extraction d'un élément d'information suivant,
acquérir une indication d'une vitesse disponible de bande passante entrante vers l'ordinateur client (100) ;
acquérir une indication d'une vitesse disponible

de bande passante sortante vers l'ordinateur serveur (150) ;
déterminer une vitesse minimale parmi la bande passante entrante disponible et la bande passante sortante disponible ; et
accepter les données associées à l'élément d'information suivant à une vitesse correspondant à une augmentation par rapport à la vitesse minimale.

4. Procédé selon la revendication 3, dans lequel l'augmentation est de 10%.

5. Procédé selon la revendication 3, dans lequel la bande passante entrante disponible est une bande passante entrante totale moins la somme du nombre de bits reçus à partir d'éléments d'information d'un rang plus élevé tournant pendant une période de temps précédente.

6. Procédé selon la revendication 5, dans lequel la bande passante entrante totale est, dans une certaine période de temps, définie par,
la détermination d'un nombre de bits reçus en provenance d'une source quelconque extérieure à l'ordinateur client (100) ;
la comparaison du nombre déterminé de bits avec la quantité de bits reçus sur des périodes de temps autres que ladite période de temps ; et
le choix du nombre de bits le plus grand déterminé à partir de la comparaison comme bande passante entrante totale.

7. Procédé selon la revendication 3 ou la revendication 4, dans lequel la bande passante disponible pour l'ordinateur serveur (150) stockant l'élément d'information suivant est la bande passante totale pour l'ordinateur serveur moins la somme du nombre de bits reçus par l'ordinateur client (100), pendant une période, à compter de l'extraction à partir de l'ordinateur serveur d'un élément d'information d'un rang plus élevé que l'élément d'information suivant.

8. Procédé selon l'une quelconque des revendications précédentes, comprenant de plus l'étape consistant à agir sur chaque élément d'information après son extraction.

9. Procédé selon la revendication 8, dans lequel l'étape consistant à agir est l'une des étapes suivantes :

afficher un texte, afficher un élément d'interface utilisateur graphique, afficher un fichier d'image, jouer un fichier audio, exécuter un mini-programme (applet), et exécuter une pluralité de codes informatiques.

10. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'élément d'information est l'un des éléments suivants : un texte, un élément d'interface utilisateur graphique, un fichier d'image, et un fichier audio, un mini-programme, et une pluralité de codes informatiques.

11. Produit de programme informatique pour extraire des éléments d'information, dans un réseau informatique (10) comprenant un ordinateur client (100) et un ordinateur serveur (150), le produit de programme informatique comprenant un code, résidant sur un support réel, code qui, quand il est exécuté, induit un processeur (102) à :

construire une liste de références d'élément d'information, au moins une des références d'élément d'information possédant un niveau de priorité associé à l'élément d'information correspondant ;
trier la liste de références d'élément d'information, de manière que la liste soit rangée d'une première référence d'élément d'information possédant la plus haute priorité à une dernière référence d'élément d'information possédant la plus basse priorité ; et
extraire, de l'ordinateur serveur vers l'ordinateur client, les éléments d'information référencés dans l'ordre de rangement.

12. Dispositif pour extraire des éléments d'information, dans un réseau informatique (10), le réseau informatique comprenant un ordinateur client (100) et un ordinateur serveur (150), le dispositif comprenant :

un mécanisme configuré pour construire une liste de références d'élément d'information, au moins une des références d'élément d'information possédant un niveau de priorité associé à l'élément d'information correspondant ;
un mécanisme configuré pour trier une liste de références d'élément d'information, de manière que la liste soit rangée d'une première référence d'élément d'information possédant la plus haute priorité à une dernière référence d'élément d'information possédant la plus basse priorité ; et
un mécanisme configuré pour extraire, de l'ordinateur serveur vers l'ordinateur client, les éléments d'information référencés dans l'ordre de rangement.

13. Dispositif selon la revendication 12, dans lequel les mécanismes configurés pour construire et trier la liste de références d'élément d'information comprennent de plus :

un mécanisme configuré pour recevoir un fichier (110) qui stocke les références d'élément d'information ;
un mécanisme configuré pour ordonner les références d'élément d'information en une liste initiale selon l'apparition séquentielle dans le fichier de la référence d'élément d'information, l'apparition séquentielle étant associée à un ou plusieurs numéros de séquence ; et
un mécanisme configuré pour réordonner la liste initiale par niveau de priorité décroissant comme clé de tri primaire et par numéro de séquence croissant comme clé de tri secondaire.

14. Dispositif selon la revendication 12 ou 13, dans lequel le mécanisme configuré pour extraire les éléments d'information comprend de plus :

un mécanisme configuré pour déterminer si au moins un élément d'information référencé dans la liste est en cours d'extraction ;
un mécanisme configuré pour faciliter l'extraction d'un élément d'information suivant référencé dans la liste, quand la détermination indique qu'au moins un élément d'information est en cours d'extraction, comprenant un mécanisme configuré pour acquérir une indication de vitesse disponible de bande passante entrante vers l'ordinateur client (100), un mécanisme configuré pour acquérir une indication de vitesse disponible de bande passante sortante vers l'ordinateur serveur (150), un mécanisme configuré pour déterminer une vitesse minimale de la bande passante entrante disponible et de la bande passante sortante disponible et un mécanisme configuré pour accepter les données associées à l'élément d'information suivant à une vitesse correspondant à une augmentation par rapport à la vitesse minimale.

15. Dispositif selon la revendication 14, dans lequel l'augmentation est de 10%.

16. Dispositif selon la revendication 14 ou 15, dans lequel la bande passante entrante disponible est une bande passante entrante totale moins la somme du nombre de bits reçus à partir d'éléments d'information d'un rang plus élevé tournant pendant une période de temps précédente.

17. Dispositif selon l'une quelconque des revendications 14, 15 ou 16, dans lequel le mécanisme configuré pour acquérir la bande passante entrante totale, dans une période de temps donnée, comprend :

un mécanisme configuré pour déterminer un nombre de bits reçus en provenance d'une source quelconque extérieure au système informatique associé au dispositif d'affichage ;
un mécanisme configuré pour comparer le nombre déterminé de bits avec la quantité de bits reçus sur des périodes de temps autres que la période de temps donnée; et
un mécanisme configuré pour choisir le nombre de bits le plus grand déterminé à partir de la comparaison comme bande passante entrante totale.

18. Dispositif selon l'une quelconque des revendications 14 à 17, dans lequel le mécanisme configuré pour acquérir la bande passante disponible vers le système informatique stockant l'élément d'information suivant comprend un mécanisme configuré pour déterminer la bande passante totale vers le système informatique stockant l'élément d'information suivant moins la somme du nombre de bits reçus, pendant une période donnée, à partir de l'extraction d'un élément d'information d'un rang plus élevé que l'élément d'information suivant.

19. Dispositif selon l'une quelconque des revendications 12 à 18, comprenant de plus un mécanisme configuré pour agir sur chaque élément d'information après son extraction.

20. Dispositif selon la revendication 19, dans lequel le mécanisme configuré pour agir réalise au moins l'une des actions suivantes : affiche un texte, ou affiche un élément d'interface utilisateur graphique, ou affiche un fichier d'image, ou joue un fichier audio, ou exécute un mini-programme, ou exécute une pluralité de codes informatiques.

21. Dispositif selon l'une quelconque des revendications 12 à 20, dans lequel l'élément d'information est l'un des éléments suivants : un texte, un élément d'interface utilisateur graphique, un fichier d'image, un fichier audio, un mini-programme, et une pluralité de codes informatiques.

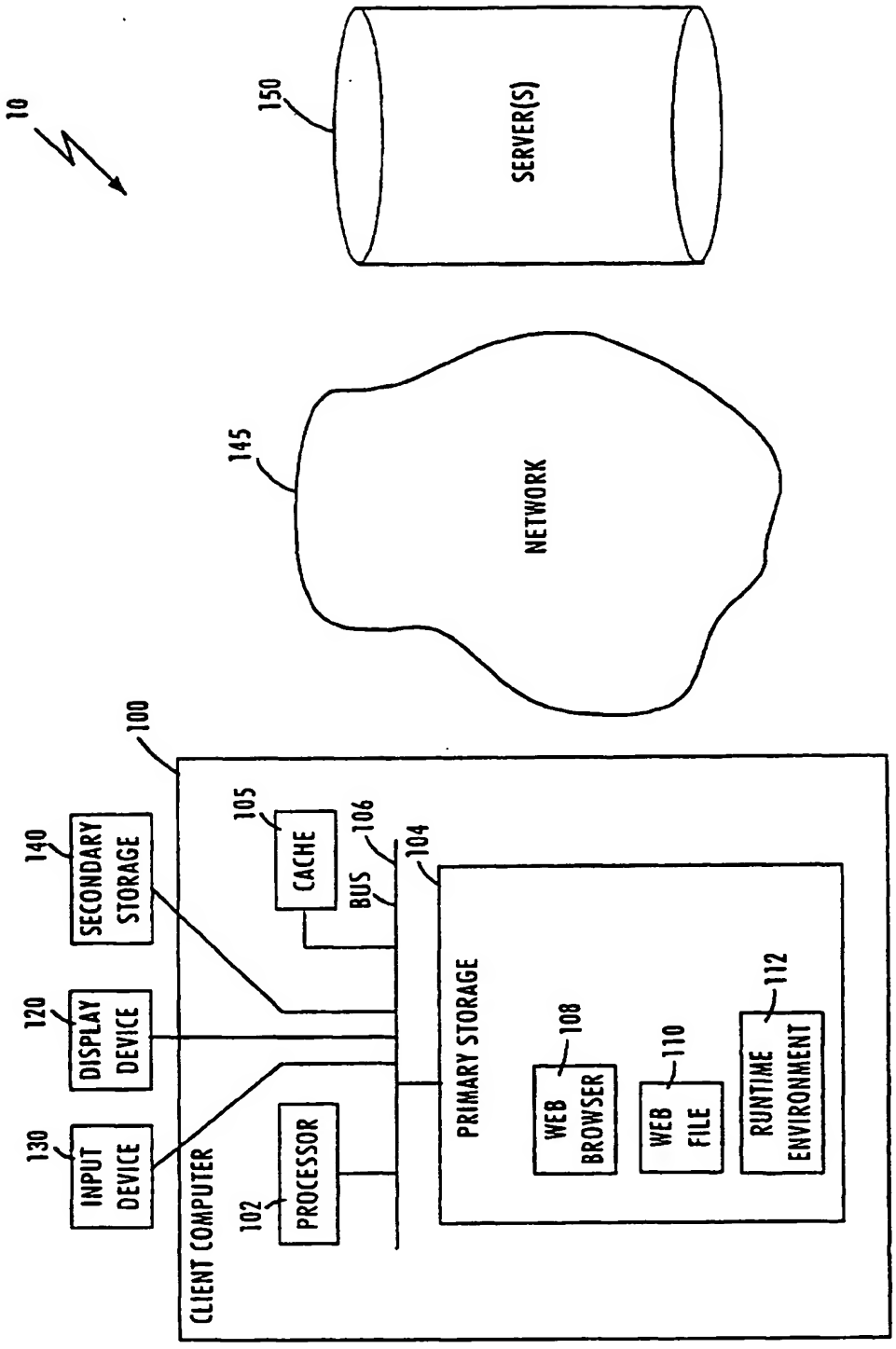

10
150
SERVER(S)
145
NETWORK
100
CLIENT COMPUTER
140
SECONDARY STORAGE
120
DISPLAY DEVICE
130
INPUT DEVICE
102
PROCESSOR
105
CACHE
106
BUS
104
PRIMARY STORAGE
108
WEB BROWSER
110
WEB FILE
112
RUNTIME ENVIRONMENT

Figure 1

```
<html>
<head>
<title>

</title>
</head>
<!-- this is a comment -->
<body>

<address>

</address>
</body>
</html>
```

Figure 2

<start tag>	<endtag>	function
<html>	</html>	HTML document indicator.
<head>	</head>	Defines document head.
<title>	</title>	Document title information. Should be descriptive, used in indexing and search engines.
<body>	</body>	Document body
<h (n)>, <h1>... <h6>	</h (n)>, </h1>... </h6>	Headings. h1 is largest, h6 smallest
<!--- --->		Comment. No ending tag required

Figure 3

THOMAS JEFFERSON WAS ONE OF THE DRAFTERS OF THE AMERICAN CONSTITUTION.

Figure 4

Url components

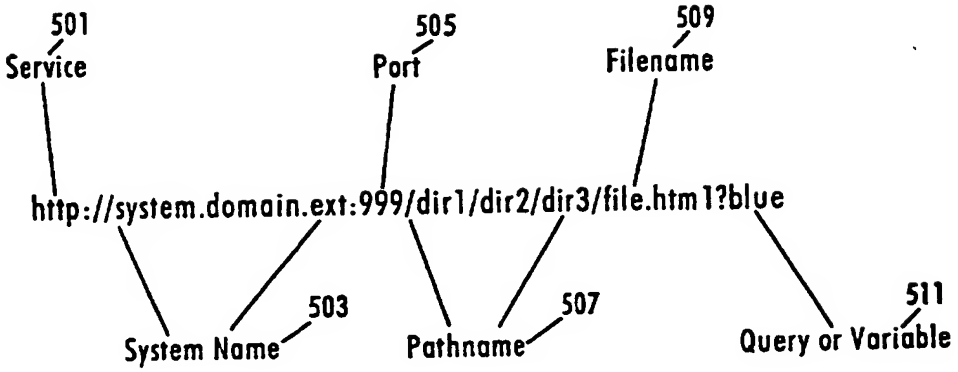


Figure 5

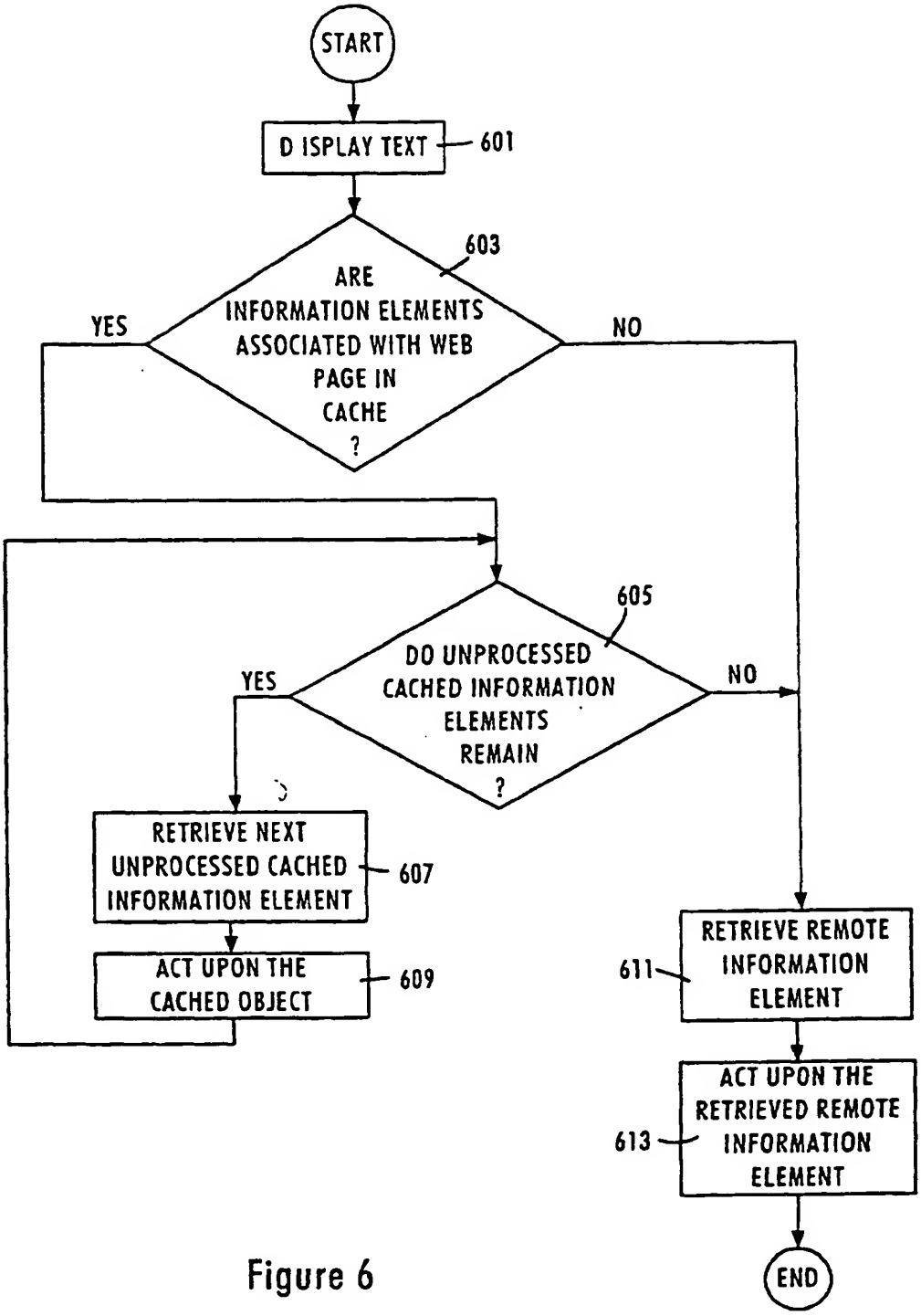

START
D ISPLAY TEXT
601
603
ARE INFORMATION ELEMENTS ASSOCIATED WITH WEB PAGE IN CACHE ?
YES
NO
605
DO UNPROCESSED CACHED INFORMATION ELEMENTS REMAIN ?
YES
NO
RETRIEVE NEXT UNPROCESSED CACHED INFORMATION ELEMENT
607
ACT UPON THE CACHED OBJECT
609
RETRIEVE REMOTE INFORMATION ELEMENT
611
ACT UPON THE RETRIEVED REMOTE INFORMATION ELEMENT
613
END

Figure 6

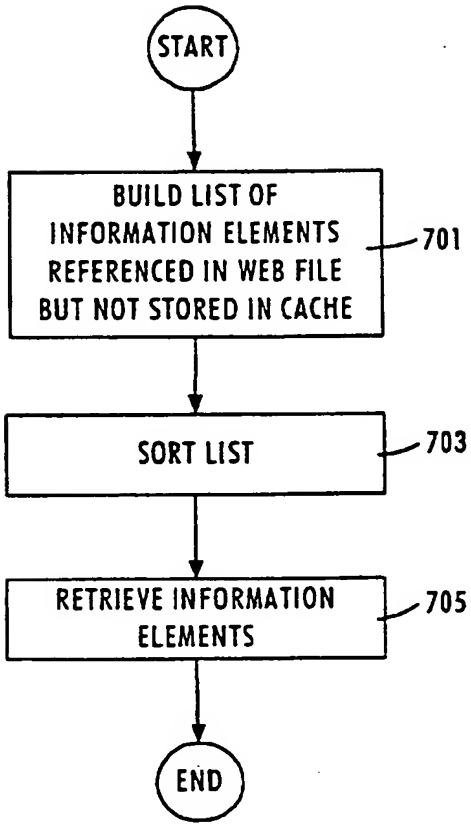
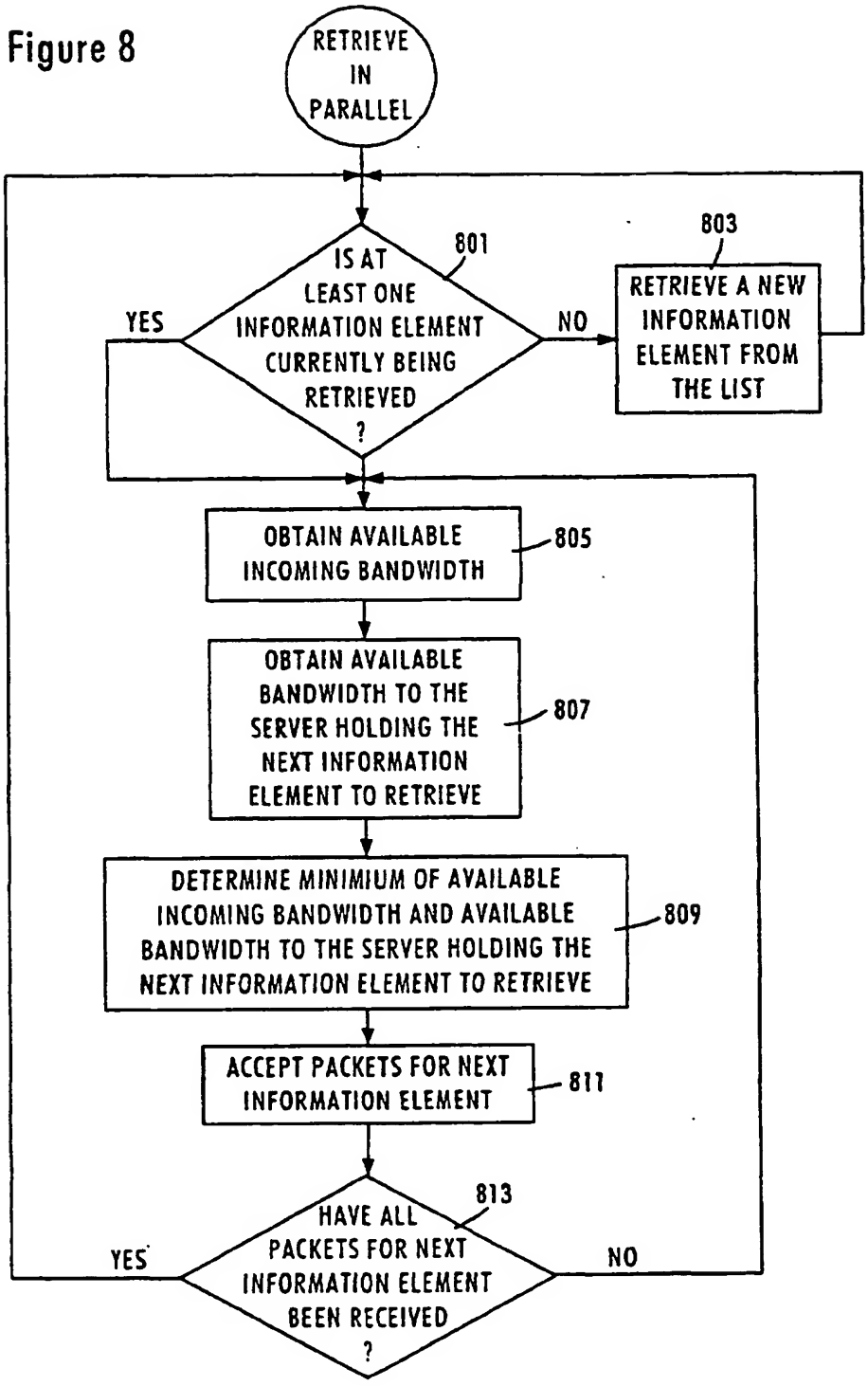

START
BUILD LIST OF INFORMATION ELEMENTS REFERENCED IN WEB FILE BUT NOT STORED IN CACHE
701
SORT LIST
703
RETRIEVE INFORMATION ELEMENTS
705
END

Figure 7

Figure 8


RETRIEVE IN PARALLEL
IS AT LEAST ONE INFORMATION ELEMENT CURRENTLY BEING RETRIEVED ?
801
YES
NO
803
RETRIEVE A NEW INFORMATION ELEMENT FROM THE LIST
OBTAIN AVAILABLE INCOMING BANDWIDTH
805
OBTAIN AVAILABLE BANDWIDTH TO THE SERVER HOLDING THE NEXT INFORMATION ELEMENT TO RETRIEVE
807
DETERMINE MINIMIUM OF AVAILABLE INCOMING BANDWIDTH AND AVAILABLE BANDWIDTH TO THE SERVER HOLDING THE NEXT INFORMATION ELEMENT TO RETRIEVE
809
ACCEPT PACKETS FOR NEXT INFORMATION ELEMENT
811
HAVE ALL PACKETS FOR NEXT INFORMATION ELEMENT BEEN RECEIVED ?
813
YES
NO